

Lab - 2

1. Explain why do we want sometimes to use `setImmediate` instead of using `setTimeout`?

`setImmediate()` schedules a callback and gets executed in the check phase of the event loop after IO event's callbacks.

`setTimeout` is to schedule a callback to run after a specific time, the functions are registered in the timers phase of the event loop.

We use `setImmediate` over `setTimeout` in the IO cycle to execute immediately before any other timers are present in the code.

2. Explain the difference between `process.nextTick` and `setImmediate`?

`process.nextTick` is not part of the event loop. `nextTick()` queue is the priority queue that runs before any additional I/O events or timers. If `process.nextTick()` is called in a given phase, all the callbacks passed to `process.nextTick()` will be resolved before the event loop continues.

`setImmediate()` is part of the event loop and it runs in the check phase, which is immediately after poll (I/O) phase.

3. Name 10 core modules that Node provides by default, and 10 of the Global objects?

Node 10 Core Modules:

- 1) http/https
- 2) path
- 3) fs
- 4) url
- 5) v8
- 6) timers
- 7) net
- 8) readline
- 9) stream
- 10) events

Node 10 Global Objects:

- 1) setTimeout
- 2) setImmediate
- 3) setInterval
- 4) clearTimeout
- 5) clearInterval
- 6) clearImmediate
- 7) process
- 8) _filename
- 9) _dirname
- 10) console

Exercise 02

```
function functorialAsync(n) {  
  return new Promise((resolve, reject) => {  
    let res = 1;  
    for (let i = n; i > 1; i--) {  
      res = res * i;  
    }  
    resolve(res)  
  })  
}
```

```
console.log('start');  
functorialAsync(500).then(console.log);  
console.log('end');
```

```
//Output  
start  
end  
Infinity
```

Observation: when we calculate the factorial of a large number, the result will show infinity and it will take longer time to complete the operation.